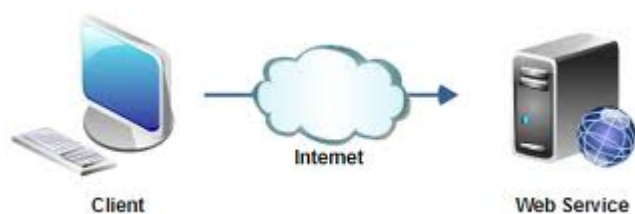


WEB SERVICES

Manual técnico
para
desarrollador



1. Introducción

Este Manual considera servir de guía para un desarrollo que deba integrarse con el Web Service que el Banco Central de Chile provee para obtener los datos (observaciones) de las series de frecuencia diaria, mensual, trimestral y anual que se dejan disponibles a través de esta tecnología.

Se debe considerar que estos ejemplos de códigos fuente están escritos para código c# y php, sin embargo llevar estos ejemplos a java u otros lenguajes no debería ser problema para el desarrollador.

Para poder comenzar a usar este Web Service la empresa deberá previamente solicitar una cuenta de acceso a través del formulario que se encuentra en:

http://si3.bcentral.cl/estadisticas/web_services/Webservices/registro.pdf

La url donde se encuentran el WSDL (Web Service Description Language) de los Web Services que el BCCh disponibiliza está en:

<https://si3.bcentral.cl/SieteWS/sietews.asmx?wsdl>

Se recomienda revisar la siguiente página web:

http://si3.bcentral.cl/estadisticas/web_services/index.htm

2. Referencia de servicio web y obtención de datos

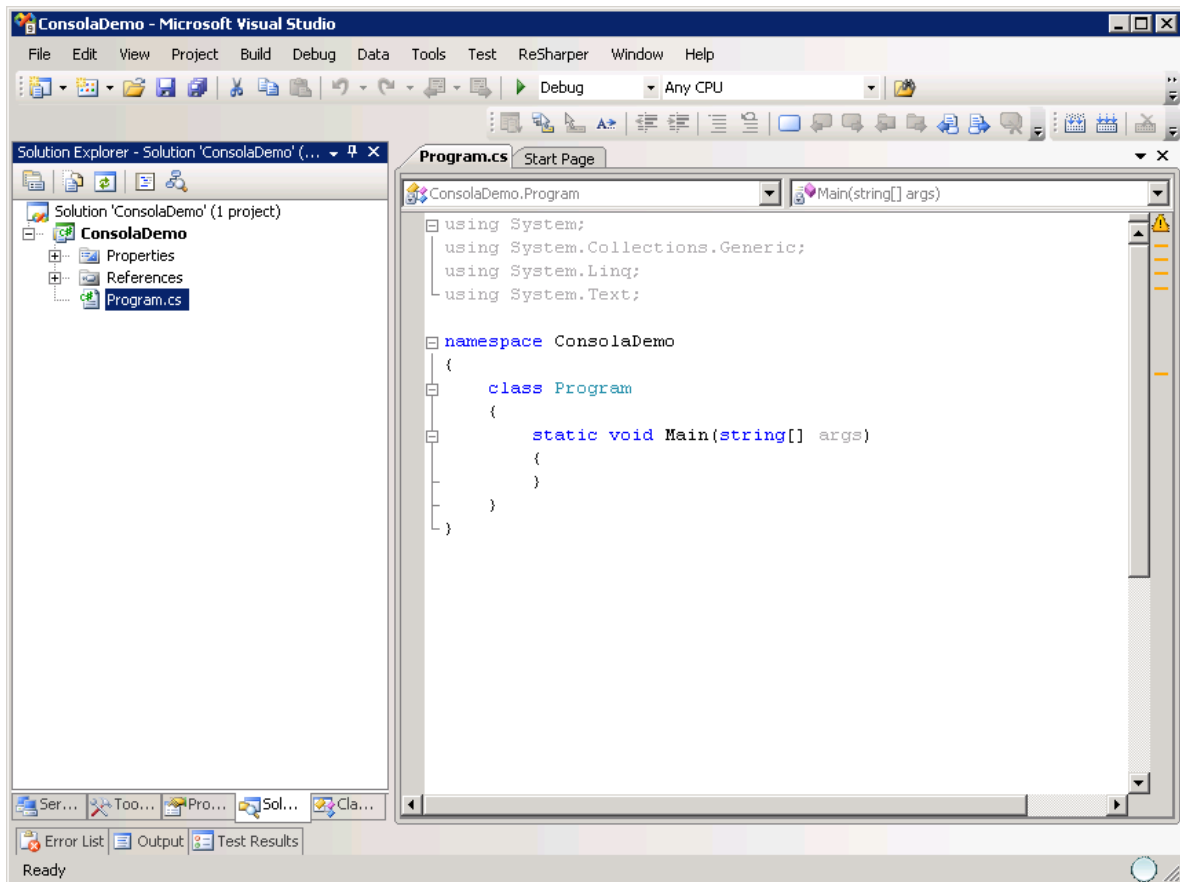
El código y la aplicación de muestra a continuación está construida utilizando visual studio 2008 y c# como lenguaje.

En el siguiente capítulo se desplegará el código tanto para c# como para php. Ambos realizan las mismas operaciones, salvo que el primero se ejecuta en una aplicación de consola y el segundo en una página web.

Para los desarrollos en PHP, los puntos 3.1 y 3.2 pueden ser omitidos.

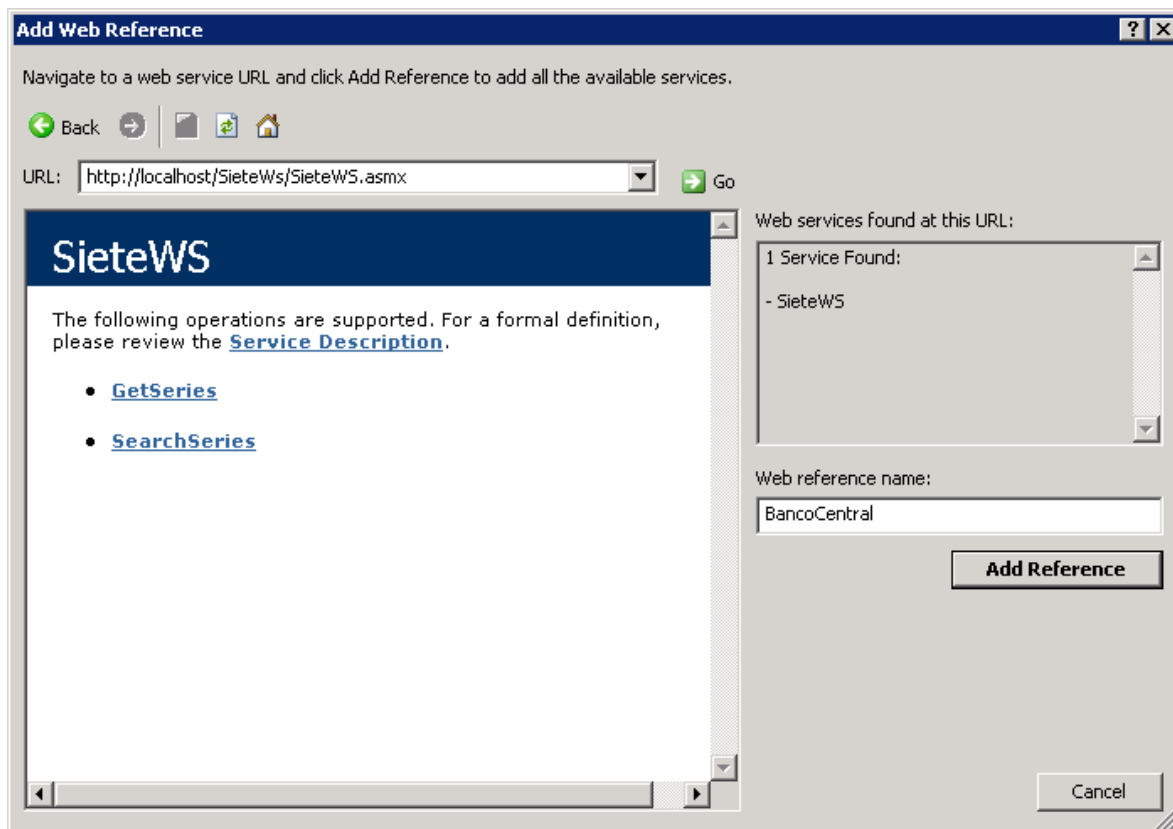
2.1 Creación de aplicación

Para llevar a cabo este paso a paso se debe contar con un proyecto .net ya creado en una solución. En este caso se utilizará una aplicación de consola.



2.2 Agregar referencia al servicio web

Con la aplicación creada, se deberá agregar una referencia al servicio web disponible en la ruta <RUTA>.



Una vez que visual studio haya terminado de agregar la referencia y creado las clases necesarias, estas podrán ser utilizadas desde la aplicación de consola.

Se debe considerar para efectos de este ejemplo que la referencia se creó con el nombre de BancoCentral.

2.3 Obtener las series disponibles

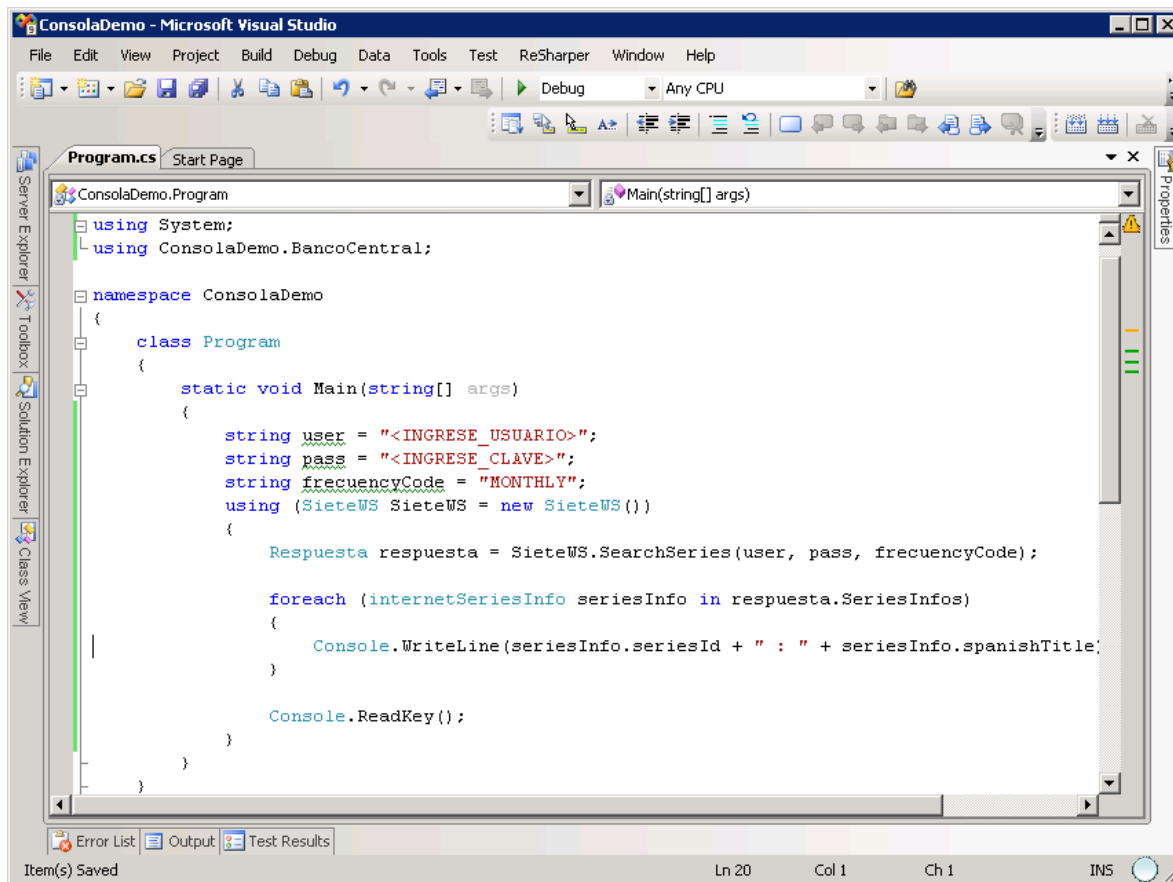
Para obtener las series disponibles se dispone de un método SearchSeries, el cual recibe tres parámetros. Estos parámetros son:

- user: nombre de usuario entregado por el banco para acceder al sistema
- password: contraseña del usuario
- frequencyCode: filtro para acotar a el resultado sólo a las series que tienen esa frecuencia

Las opciones válidas de frequencyCode se listan a continuación:

- DAILY
- MONTHLY
- QUARTERLY
- ANNUAL

La siguiente imagen muestra el bloque de código necesario para obtener las series que tienen datos con la frecuencia seleccionada (MONTHLY para el ejemplo).



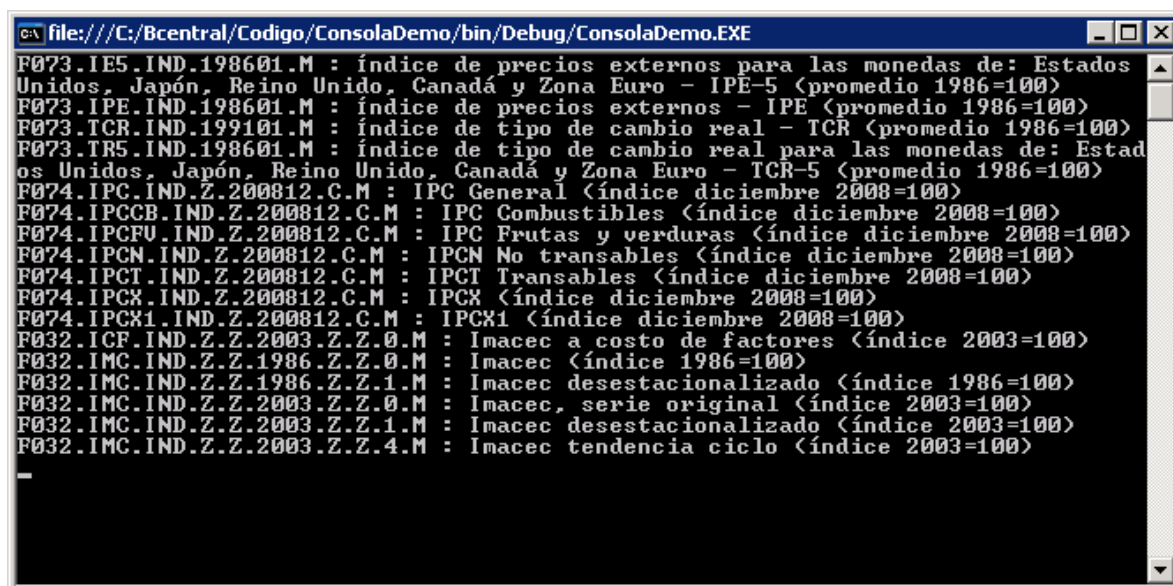
```
using System;
using ConsolaDemo.BancoCentral;

namespace ConsolaDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            string user = "<INGRESE_USUARIO>";
            string pass = "<INGRESE_CLAVE>";
            string frequencyCode = "MONTHLY";
            using (SieteWS SieteWS = new SieteWS())
            {
                Respuesta respuesta = SieteWS.SearchSeries(user, pass, frequencyCode);

                foreach (internetSeriesInfo seriesInfo in respuesta.SeriesInfos)
                {
                    Console.WriteLine(seriesInfo.seriesId + " : " + seriesInfo.spanishTitle);
                }

                Console.ReadKey();
            }
        }
    }
}
```

La ejecución de este código despliega el siguiente listado de series. Además del id de la serie y el nombre en español que son desplegados, también se obtiene información de las fechas que tienen observaciones para cada serie.



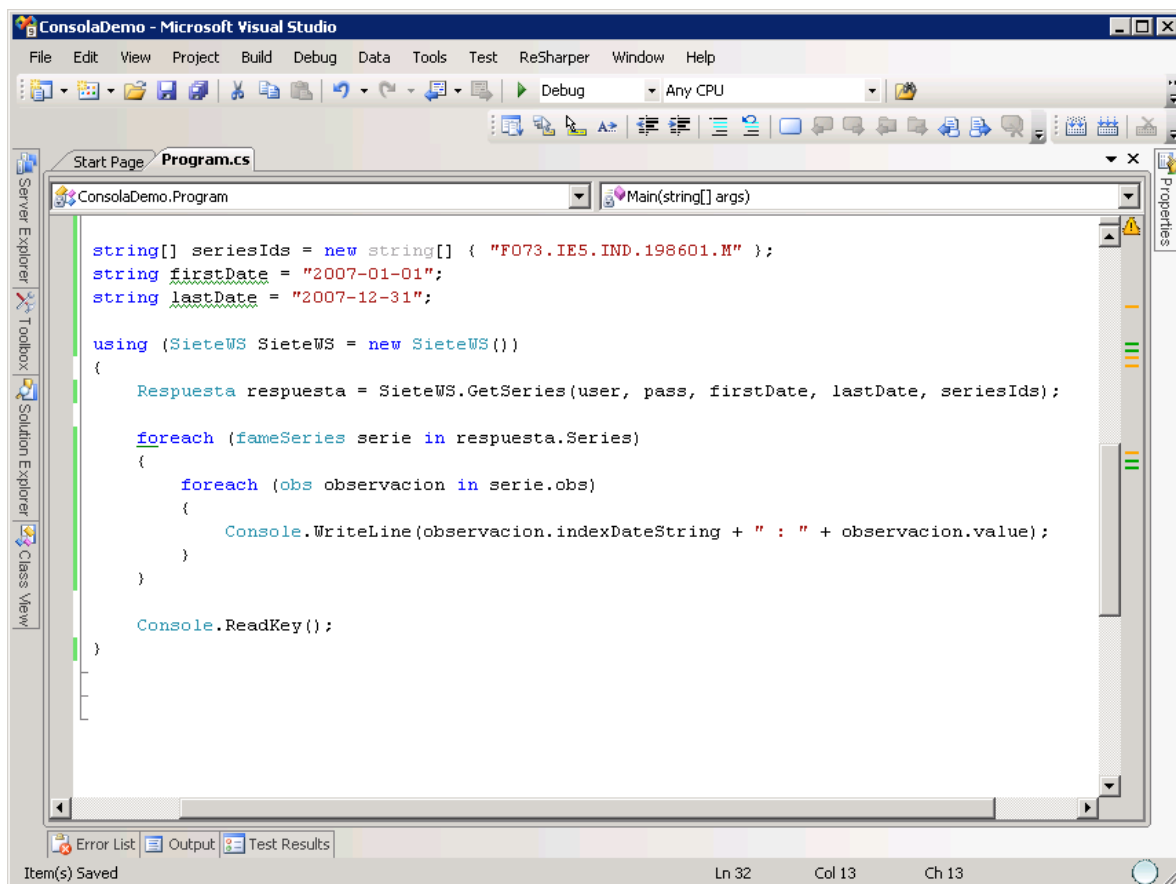
```
file:///C:/Bcentral/Codigo/ConsolaDemo/bin/Debug/ConsolaDemo.EXE
F073.IE5.IND.198601.M : índice de precios externos para las monedas de: Estados Unidos, Japón, Reino Unido, Canadá y Zona Euro - IPE-5 (promedio 1986=100)
F073.IPE.IND.198601.M : índice de precios externos - IPE (promedio 1986=100)
F073.TCR.IND.199101.M : índice de tipo de cambio real - TCR (promedio 1986=100)
F073.TR5.IND.198601.M : índice de tipo de cambio real para las monedas de: Estados Unidos, Japón, Reino Unido, Canadá y Zona Euro - TCR-5 (promedio 1986=100)
F074.IPC.IND.Z.200812.C.M : IPC General (índice diciembre 2008=100)
F074.IPCCB.IND.Z.200812.C.M : IPC Combustibles (índice diciembre 2008=100)
F074.IPCFU.IND.Z.200812.C.M : IPC Frutas y verduras (índice diciembre 2008=100)
F074.IPCN.IND.Z.200812.C.M : IPCN No transables (índice diciembre 2008=100)
F074.IPCT.IND.Z.200812.C.M : IPCT Transables (índice diciembre 2008=100)
F074.IPCX.IND.Z.200812.C.M : IPCX (índice diciembre 2008=100)
F074.IPCX1.IND.Z.200812.C.M : IPCX1 (índice diciembre 2008=100)
F032.ICF.IND.Z.Z.2003.Z.Z.0.M : Imacec a costo de factores (índice 2003=100)
F032.IMC.IND.Z.Z.1986.Z.Z.0.M : Imacec (índice 1986=100)
F032.IMC.IND.Z.Z.1986.Z.Z.1.M : Imacec desestacionalizado (índice 1986=100)
F032.IMC.IND.Z.Z.2003.Z.Z.0.M : Imacec, serie original (índice 2003=100)
F032.IMC.IND.Z.Z.2003.Z.Z.1.M : Imacec desestacionalizado (índice 2003=100)
F032.IMC.IND.Z.Z.2003.Z.Z.4.M : Imacec tendencia ciclo (índice 2003=100)
```

2.4 Obtener las observaciones para una serie

Para obtener las observaciones para una serie se dispone de un método llamado `GetSeries`, el cual recibe cinco parámetros. Estos parámetros son:

- `user`: nombre de usuario entregado por el banco para acceder al sistema
- `password`: contraseña del usuario
- `firstDate`: filtro para acotar las observaciones a partir de una fecha de inicio. El parámetro es opcional. El formato debe ser YYYY-MM-DD
- `lastDate`: filtro para acotar las observaciones hasta una fecha de término. El parámetro es opcional. El formato debe ser YYYY-MM-DD
- `seriesIds`: filtro para indicar la serie de la que se quieren obtener las observaciones. El parámetro es obligatorio.

Por ejemplo, si se desean obtener las observaciones para la primera serie del listado de más arriba, se debe utilizar el ID de ésta, el cual corresponde a "F073.IE5.IND.198601.M". Si se quieren obtener todas las del año 2007, se deberá llamar el servicio web como se muestra en el código de más abajo.



```
string[] seriesIds = new string[] { "F073.IE5.IND.198601.M" };
string firstDate = "2007-01-01";
string lastDate = "2007-12-31";

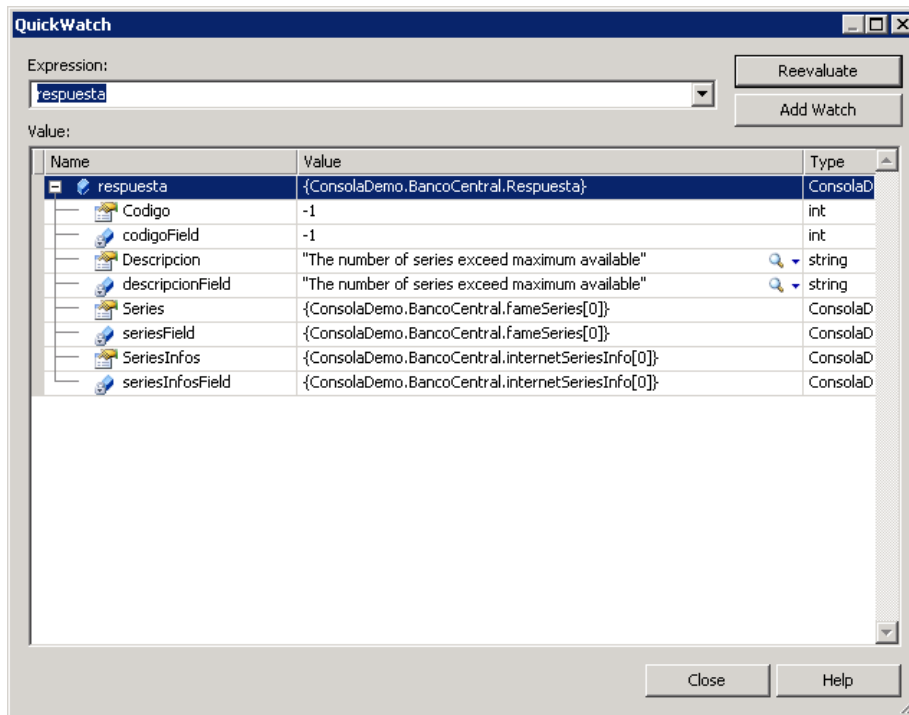
using (SieteWS SieteWS = new SieteWS())
{
    Respuesta respuesta = SieteWS.GetSeries(user, pass, firstDate, lastDate, seriesIds);

    foreach (fameSeries serie in respuesta.Series)
    {
        foreach (obs observacion in serie.obs)
        {
            Console.WriteLine(observacion.indexDateString + " : " + observacion.value);
        }
    }

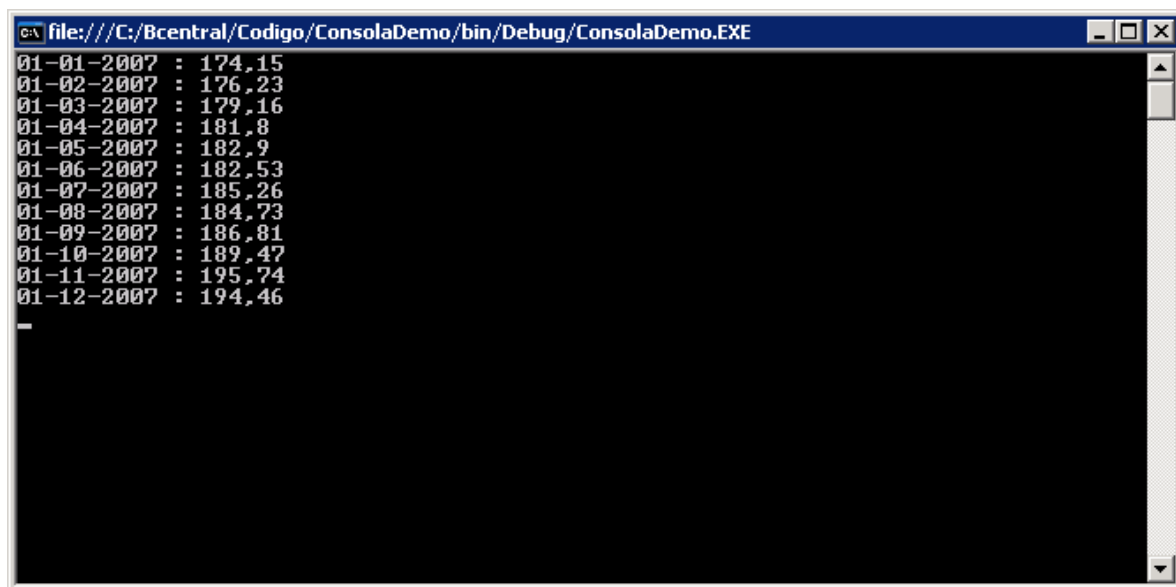
    Console.ReadKey();
}
```

En el código se observa que es necesario hacer un ciclo `foreach` dentro de otro ciclo `foreach`. El primer ciclo recorre selecciona la serie solicitada y el segundo ciclo recorre las observaciones de esta serie.

Existe un máximo de 1 serie permitido para el cual se pueden obtener las mediciones. Un mensaje será retornado en la respuesta indicando que se solicitó más de una serie.



La ejecución de este código da el resultado desplegado a continuación, en donde destaca la fecha de la observación, una por mes dado que es mensual (MONTHLY), y el valor de la observación.

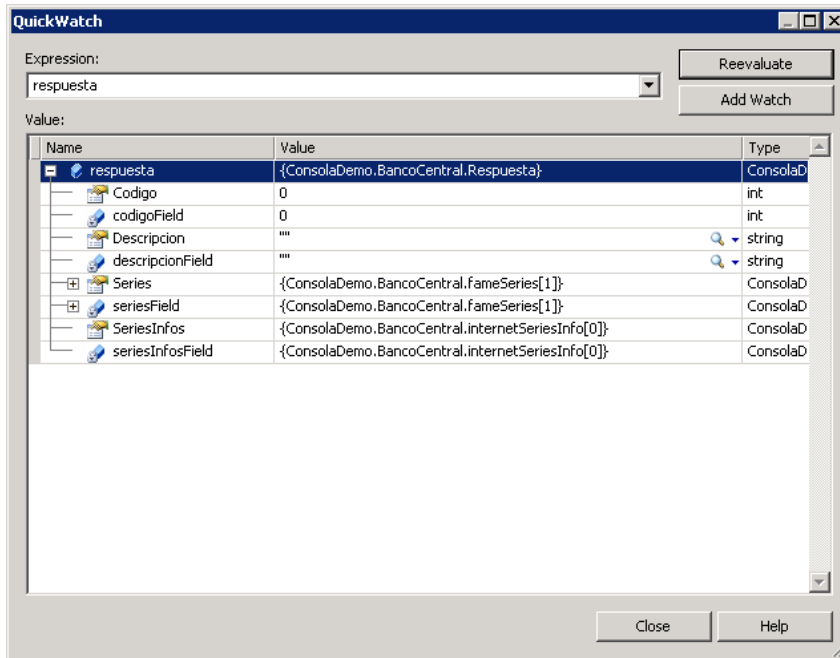


2.5 Control de errores

Como se vio en el ejemplo anterior, en caso de solicitar más series de las permitidas, se devolverá una respuesta indicándolo.

En la respuesta recibida contiene dos propiedades que indicarán el éxito o fracaso de la ejecución. Estas propiedades son Codigo y Descripcion. En caso de una ejecución sin problemas, los valores que retornarán serán cero y una cadena vacía respectivamente.

Esto se puede ver en la siguiente imagen.



3. Código de las aplicaciones

Como se mencionó en el capítulo anterior, los siguientes bloques de código cumplen funcionalidad similar, aunque los ambientes de ejecución son diferentes.

3.1 Código de aplicación de consola en c#

```
using System;
using ConsolaDemo.BancoCentral;

namespace ConsolaDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            string user = "<INGRESE_USUARIO>";
            string pass = "<INGRESE_CLAVE>";
            string frecuencyCode = "MONTHLY";
            using (SieteWS SieteWS = new SieteWS())
            {
                Respuesta respuesta = SieteWS.SearchSeries(user, pass, frecuencyCode);

                foreach (internetSeriesInfo seriesInfo in respuesta.SeriesInfos)
                {
                    Console.WriteLine(seriesInfo.seriesId + " : " + seriesInfo.spanishTitle);
                }

                Console.ReadKey();
            }

            string[] seriesIds = new string[] { "F073.IE5.IND.198601.M" };
            string firstDate = "2007-01-01";
            string lastDate = "2007-12-31";

            using (SieteWS SieteWS = new SieteWS())
            {
                Respuesta respuesta = SieteWS.GetSeries(user, pass, firstDate, lastDate, seriesIds);

                foreach (fameSeries serie in respuesta.Series)
                {
                    foreach (obs observacion in serie.obs)
                    {
                        Console.WriteLine(observacion.indexDateString + " : " + observacion.value);
                    }
                }

                Console.ReadKey();
            }
        }
    }
}
```

3.2 Código de página web en php

```
<?
$user = '<INGRESE_USUARIO>';
$password='<INGRESE_CLAVE>';
$frequencyCode ='MONTHLY';

$wsdl="<DIRECCION_SITIO_WEB>?WSDL";

$client = new soapclient($wsdl);
$params = new stdClass;
$params->user = $user;
$params->password = $password;
$params->frequencyCode = $frequencyCode;
$result = $client->SearchSeries($params)->SearchSeriesResult;

foreach ($result->SeriesInfos->internetSeriesInfo as $serie)
{
    echo $serie->seriesId . " : " . $serie->spanishTitle . "<br/>";
}

$seriesIds = array ("F073.IE5.IND.198601.M");
$firstDate = "2007-01-01";
$lastDate = "2007-12-31";

$client = new soapclient($wsdl);
$params = new stdClass;
$params->user = $user;
$params->password = $password;
$params->firstDate = $firstDate;
$params->lastDate = $lastDate;
$params->seriesIds = $seriesIds;

$result = $client->GetSeries($params)->GetSeriesResult;

$fameSeries =$result->Series->fameSeries;

//Cuando se solicita una sola serie, la respuesta no es interpretada como un arreglo
if (is_array($fameSeries ) != 1) $fameSeries = array($fameSeries);

foreach ($fameSeries as $serieFame)
{
    echo $serieFame->seriesKey->seriesId . " : " . $serieFame->precision . "<br/>";
    foreach ($serieFame->obs as $obs)
    {
        echo $obs->indexDateString . " : " . $obs->value . "<br/>";
    }
}
?>
```
